Aggregation of estimators

Olena Verbytska

1 Introduction

An aggregated estimator is a convex or linear combination of estimators. This choice of an estimator allows to preserve all the good properties of each estimator. Such model is less sensitive to small perturbations of the data Y.

All the results are taken from the book [1], Sections 2 and 4. The main result of this paper is Theorem 3.3 about oracle risk bound for the Gibbs mixing estimator which will be compared to oracle risk bound for model selection. Then we consider the numerical approximation of aggregated estimator by Metropolis-Hastings algorithm. And finally we will simulate Gaussian white noise data and apply the theory to recreate the unknown signal.

2 Definitions

Let us consider a set of estimators $\{\widehat{f}_m, m \in \mathscr{M}\}$. Instead of selecting only one estimator $\widehat{f}_{\widehat{m}}$, we will consider a convex combination $\widehat{f}$ of the $\{\widehat{f}_m\}_m$

$$\widehat{f} = \sum_{m\in\mathscr{M}} w_m \widehat{f}_m, \quad \text{with} \quad w_m \geq 0 \quad \text{and} \quad \sum_{m\in\mathscr{M}} w_m = 1.$$

Now we consider a collection of models $\{S_m, m \in \mathscr{M}\}$, where $\{S_m\}_m$ are linear subspaces of $\mathbb{R}^n$. The estimator $\widehat{f}_m = \mathrm{Proj}_{S_m} Y$ maximizes the likelihood under the constraint that it belongs to S_m. Then the unbiased estimator of the risk $r_m = \mathbb{E}\big[\|f^* - \widehat{f}_m\|^2\big]$ is

$$\widehat{r}_m = \|Y - \widehat{f}_m\|^2 + 2d_m\sigma^2 - n\sigma^2, \quad \text{with } d_m = \dim(S_m).$$

We associate to $\beta > 0$ and a probability distribution π on $\mathscr{M}$ the Gibbs mixing $\widehat{f}$ of the collection of estimators $\{\widehat{f}_m\}_m$

$$\widehat{f} = \sum_{m\in\mathscr{M}} w_m \widehat{f}_m, \quad \text{with } w_m = \frac{\pi_m e^{-\beta\widehat{r}_m/\sigma^2}}{\mathscr{Z}}, \quad \text{where } \mathscr{Z} = \sum_{m\in\mathscr{M}} \pi_m e^{-\beta\widehat{r}_m/\sigma^2}. \tag{1}$$

The Gibbs distribution w corresponds to the distribution *minimizing* over the set of probability q on $\mathscr{M}$ the functional

$$\mathscr{G}(q) = \sum_{m\in\mathscr{M}} q_m\widehat{r}_m + \frac{\sigma^2}{\beta}\mathscr{K}(q,\pi), \tag{2}$$

where $\mathscr{K}(q,\pi) = \sum_{m\in\mathscr{M}} q_m \log(q_m/\pi_m) \geq 0$ is the Kullback-Leibler divergence between the probabilities q and π.

3 Theoretical results

Proposition 3.1 (**Stein's formula**). *Let Y be a n-dimensional Gaussian vector with mean μ and covariance matrix $\sigma^2 I_n$. For any function $F : \mathbb{R}^n \to \mathbb{R}^n, F(x) = (F_1(x), \dots, F_n(x))$ continuous, with piecewise continuous partial derivatives, fulfilling for all $i \in \{1, \dots, n\}$*

(a) $\lim_{|y_i| \to \infty} F_i(y_1, \dots, y_n) e^{-(y_i - \mu_i)^2/2\sigma^2} = 0$ *for all* $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n) \in \mathbb{R}^{n-1}$,

(b) $\mathbb{E}\left[|\partial_i F_i(Y)|\right] < \infty$,

we have

$$\mathbb{E}\left[\|F(Y) - \mu\|^2\right] = \mathbb{E}[\|F(Y) - Y\|^2 - n\sigma^2 + 2\sigma^2 \operatorname{div}(F)(Y)],$$

where $\operatorname{div}(F) = \sum_i \partial_i F_i$.

Proof.

$$\begin{aligned}\mathbb{E}\left[\|F(Y) - \mu\|^2\right] &= \mathbb{E}\left[\|F(Y) - Y\|^2\right] + \underbrace{\mathbb{E}\left[\|Y - \mu\|^2\right]}_{(2)} + 2(\underbrace{\mathbb{E}\left[\langle F(Y), Y - \mu\rangle\right]}_{(3)} - \underbrace{\mathbb{E}\left[\langle Y, Y - \mu\rangle\right]}_{(1)}) \\ &= \mathbb{E}\left[\|F(Y) - Y\|^2\right] + n\sigma^2 + 2\sigma^2 \mathbb{E}[\operatorname{div}(F)(Y)] - 2n\sigma^2.\end{aligned}$$

Since Y is a Gaussian vector with mean μ and covariance matrix $\sigma^2 I_n$, we have

(1) $\mathbb{E}\left[\langle Y, Y - \mu\rangle\right] = \mathbb{E}\left[\sum_i (Y_i^2 - Y_i \mu_i)\right] = \sum_i (\mathbb{E}\left[Y_i^2\right] - \mathbb{E}\left[Y_i\right] \mu_i) = \sum_i \operatorname{Var} Y_i = n\sigma^2$,

(2) $\mathbb{E}\left[\|Y - \mu\|^2\right] = \sum_i (\mathbb{E}\left[Y_i^2\right] - 2\mathbb{E}\left[Y_i\right] \mu_i + \mu_i^2) = \sum_i (\mathbb{E}\left[Y_i^2\right] - (\mathbb{E}\left[Y_i\right])^2) = \sum_i \operatorname{Var} Y_i = n\sigma^2$.

An integration by parts, (a) and (b) give

$$\sigma^2 \int_{y_i \in \mathbb{R}} \partial_i F_i\left(y_1, \dots, y_n\right) e^{-(y_i - \mu_i)^2/2\sigma^2} dy_i = \int_{y_i \in \mathbb{R}} \left(y_i - \mu_i\right) F_i\left(y_1, \dots, y_n\right) e^{-(y_i - \mu_i)^2/2\sigma^2} dy_i,$$

for all $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n) \in \mathbb{R}^{n-1}$. Integrating with respect to $y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n$, it follows that $\mathbb{E}\left[\left(Y_i - \mu_i\right) F_i(Y)\right] = \sigma^2 \mathbb{E}\left[\partial_i F_i(Y)\right]$ and finally we get
(3) $\mathbb{E}[\langle F(Y), Y - \mu\rangle] = \sigma^2 \mathbb{E}[\operatorname{div}(F)(Y)]$.

□

Now we will compare the oracle risk bounds for model selection and for the Gibbs mixing estimator. In the first case we choose $\widehat{m}$ such that for $K > 1$

$$\widehat{m} \in \operatorname*{argmin}_{m \in \mathscr{M}} \left\{ \left\| Y - \widehat{f}_m \right\|^2 + \sigma^2 \operatorname{pen}(m) \right\}, \text{ with } \operatorname{pen}(m) = K \left(\sqrt{d_m} + \sqrt{2 \log\left(1/\pi_m\right)} \right)^2. \tag{3}$$

Theorem 3.2 (**Oracle risk bound for model selection**). *For $\widehat{f} = \widehat{f}_{\widehat{m}}$ there exists a constant $C_K > 1$ depending only on $K > 1$, such that*

$$\mathbb{E}\left[\left\| \widehat{f} - f^* \right\|^2 \right] \leq C_K \min_{m \in \mathscr{M}} \left\{ \mathbb{E}\left[\left\| \widehat{f}_m - f^* \right\|^2 \right] + \sigma^2 \log\left(\frac{1}{\pi_m} \right) + \sigma^2 \right\}. \tag{4}$$

Theorem 3.3 (**Oracle risk bound for the Gibbs mixing estimator**). *For the Gibbs mixing $\widehat{f}$ and $\beta \leq 1/4$ we have*

$$\mathbb{E}\left[\left\| \widehat{f} - f^* \right\|^2 \right] \leq \min_{m \in \mathscr{M}} \left\{ \mathbb{E}\left[\left\| \widehat{f}_m - f^* \right\|^2 \right] + \frac{\sigma^2}{\beta} \log\left(\frac{1}{\pi_m} \right) \right\}. \tag{5}$$

The risk bounds are very similar. The main difference is that the constant $C_K > 1$ in (4) and $C_K = 1$ in (5). Thus, the Gibbs mixing estimator has better oracle risk bound.

Proof of Theorem 3.3. We apply the Stein's formula to $F(Y) = \widehat{f}$ and use the linearity of the divergence operator:

$$\widehat{r} = \|Y - \widehat{f}\|^2 - n\sigma^2 + 2\sigma^2 \sum_{m \in \mathscr{M}} \operatorname{div}\left(w_m \widehat{f}_m\right) \text{ fulfills } \mathbb{E}[\widehat{r}] = \mathbb{E}\left[\left\|\widehat{f} - f^*\right\|^2\right].$$

After a bounding of $\widehat{r}$, the result will follow from the fact that w minimizes (2).

1. Upper bound of $\widehat{r}$. We first expand

$$\begin{aligned}
\|Y - \widehat{f}\|^2 &= \left\langle \sum_{m \in \mathscr{M}} w_m \left(Y - \widehat{f}_m\right), Y - \widehat{f}_m + \widehat{f}_m - \widehat{f} \right\rangle \\
&= \sum_{m \in \mathscr{M}} w_m \left\langle Y - \widehat{f}_m, Y - \widehat{f}_m \right\rangle + \sum_{m \in \mathscr{M}} w_m \left\langle Y - \widehat{f_m}, \widehat{f_m} - \widehat{f} \right\rangle \\
&= \sum_{m \in \mathscr{M}} w_m \left\|Y - \widehat{f}_m\right\|^2 + \underbrace{\sum_{m \in \mathscr{M}} w_m \left\langle Y - \widehat{f}, \widehat{f}_m - \widehat{f} \right\rangle}_{=0} - \sum_{m \in \mathscr{M}} w_m \left\|\widehat{f} - \widehat{f}_m\right\|^2,
\end{aligned}$$

and notice that the divergence of $w_m \widehat{f}_m$ is given by

$$\operatorname{div}\left(w_m \widehat{f}_m\right) = w_m \operatorname{div}\left(\widehat{f}_m\right) + \left\langle \widehat{f}_m, \nabla w_m \right\rangle.$$

Plugging these two formulas in the definition of $\widehat{r}$ we obtain

$$\widehat{r} = \sum_{m \in \mathscr{M}} w_m \left(\left\|Y - \widehat{f}_m\right\|^2 - n\sigma^2 + 2\sigma^2 \operatorname{div}\left(\widehat{f}_m\right)\right) + \sum_{m \in \mathscr{M}} w_m \left(2\sigma^2 \left\langle \widehat{f}_m, \frac{\nabla w_m}{w_m} \right\rangle - \left\|\widehat{f} - \widehat{f}_m\right\|^2\right).$$

For a linear function $G(Y) = AY$, we have $\partial_i G_i(Y) = A_{ii}$, so $\operatorname{div}(G)(Y) = \operatorname{Tr}(A)$. Since $\widehat{f}_m$ is defined as an orthogonal projection $\operatorname{Proj}_{S_m}(Y)$, we have $\operatorname{div}(\widehat{f}_m) = \operatorname{Tr}(\operatorname{Proj}_{S_m}) = d_m$ and then

$$\left\|Y - \widehat{f}_m\right\|^2 - n\sigma^2 + 2\sigma^2 \operatorname{div}\left(\widehat{f}_m\right) = \left\|Y - \widehat{f}_m\right\|^2 - n\sigma^2 + 2d_m\sigma^2 = \widehat{r}_m.$$

In addition, we have

$$\begin{aligned}
\frac{\nabla w_m}{w_m} &= -2\frac{\beta}{\sigma^2}\left(Y - \widehat{f}_m\right) - \frac{\nabla \mathscr{Z}}{\mathscr{Z}} = -2\frac{\beta}{\sigma^2}\left(\left(Y - \widehat{f}_m\right) - \sum_{m' \in \mathscr{M}} w_{m'}\left(Y - \widehat{f}_{m'}\right)\right) \\
&= -2\frac{\beta}{\sigma^2}\left(\widehat{f} - \widehat{f}_m\right),
\end{aligned}$$

$$
\begin{aligned}
\widehat{r} &= \sum_{m\in\mathscr{M}} w_m \widehat{r}_m - 4\beta \sum_{m\in\mathscr{M}} w_m \left\langle \widehat{f}_m, \widehat{f} - \widehat{f}_m \right\rangle - \sum_{m\in\mathscr{M}} w_m \left\| \widehat{f} - \widehat{f}_m \right\|^2 \\
&= \sum_{m\in\mathscr{M}} w_m \widehat{r}_m - 4\beta \Bigg(\sum_{m\in\mathscr{M}} w_m \left\langle \widehat{f}_m - \widehat{f}, \widehat{f} - \widehat{f}_m \right\rangle + \underbrace{\sum_{m\in\mathscr{M}} w_m \left\langle \widehat{f}, \widehat{f} - \widehat{f}_m \right\rangle}_{=0} \Bigg) - \sum_{m\in\mathscr{M}} w_m \left\| \widehat{f} - \widehat{f}_m \right\|^2 \\
&= \sum_{m\in\mathscr{M}} w_m \widehat{r}_m + (4\beta - 1) \sum_{m\in\mathscr{M}} w_m \left\| \widehat{f} - \widehat{f}_m \right\|^2 \\
&\leq \sum_{m\in\mathscr{M}} w_m \widehat{r}_m, \text{ when } \beta \leq 1/4.
\end{aligned}
$$

2. Optimality condition. Since $\mathscr{K}(w,\pi) \geq 0$ and w minimizes $\mathscr{G}$ over the set of probability on $\mathscr{M}$, we have

$$
\widehat{r} \leq \mathscr{G}(w) \leq \mathscr{G}(q) = \sum_{m\in\mathscr{M}} q_m \widehat{r}_m + \frac{\sigma^2}{\beta} \mathscr{K}(q,\pi), \qquad \text{for any probability } q \text{ on } \mathscr{M}.
$$

Taking the expectation, we obtain

$$
\mathbb{E}\left[\left\| \widehat{f} - f^* \right\|^2 \right] = \mathbb{E}[\widehat{r}] \leq \mathbb{E}[\mathscr{G}(q)] = \sum_{m\in\mathscr{M}} q_m r_m + \frac{\sigma^2}{\beta} \mathscr{K}(q,\pi)
$$

with $r_m = \mathbb{E}\left[\widehat{r}_m\right] = \mathbb{E}\left[\left\| \widehat{f}_m - f^* \right\|^2 \right]$. The right-hand side of the above bound is minimum for $q_m = \pi_m e^{-\beta r_m/\sigma^2} / \mathscr{Z}'$ with $\mathscr{Z}' = \sum_m \pi_m e^{-\beta r_m/\sigma^2}$. Since $\mathscr{Z}' \geq \pi_m e^{-\beta r_m/\sigma^2}$ for all $m \in \mathscr{M}$ we observe that

$$
\sum_{m\in\mathscr{M}} q_m r_m + \frac{\sigma^2}{\beta} \mathscr{K}(q,\pi) = -\frac{\sigma^2}{\beta} \log\left(\mathscr{Z}'\right) \leq \min_{m\in\mathscr{M}} \left\{ r_m - \frac{\sigma^2}{\beta} \log\left(\pi_m\right) \right\}
$$

□

4 Numerical approximation by Metropolis-Hastings

We can compute an approximation of aggregated estimator (1) by using Markov Chain Monte-Carlo (MCMC) technics, which is more efficient than computing the estimator directly, especially when the cardinality of $\mathscr{M}$ is large.

Let $F : \mathscr{M} \to \mathbb{R}$ be a real function on $\mathscr{M}$ and w be an arbitrary probability distribution on $\mathscr{M}$ fulfilling $w_m > 0$ for all $m \in \mathscr{M}$. The Metropolis-Hastings algorithm is a classical tool to approximate

$$\mathbb{E}_w[F] := \sum_{m \in \mathscr{M}} w_m F(m).$$

The Metropolis-Hastings algorithm allows to generate an ergodic (i.e. aperiodic, irreducible, and positive recurrent) Markov chain $(M_t)_{t\in\mathbb{N}}$ in $\mathscr{M}$ with stationary distribution w and to approximate $\mathbb{E}_w[F]$ by the average $T^{-1}\sum_{t=1}^{T} F(M_t)$ with T large. This approximation relies on the fact that for any ergodic Markov chain $(M_t)_{t\in\mathbb{N}}$ in $\mathscr{M}$ with stationary distribution w, we have

$$\frac{1}{T}\sum_{t=1}^{T} F(M_t) \xrightarrow[\text{a.s.}]{T\to\infty} \mathbb{E}_w[F]. \tag{6}$$

An aperiodic and irreducible Markov chain which is reversible according to the probability w is ergodic. The Metropolis-Hastings algorithm proposes a way to build a Markov chain $(M_t)_{t\in\mathbb{N}}$ the numerical generation of which is computationally efficient.

Let $\Gamma(m, m')$ be the transition probability of an aperiodic irreducible random walk on $\mathscr{M}$ and define

$$Q(m, m') = \Gamma(m, m') \wedge \frac{w_{m'}\Gamma(m', m)}{w_m} \quad \text{for all } m \neq m',$$
$$Q(m, m) = 1 - \sum_{m' : m' \neq m} Q(m, m') \quad \text{for all } m \in \mathscr{M}.$$

Then we can see that Markov(w, Q) is reversible. Therefore the Markov chain $(M_t)_{t\in\mathbb{N}}$ with transition Q fulfills for any initial condition

$$\frac{1}{T}\sum_{t=1}^{T} F(M_t) \xrightarrow[\text{a.s.}]{T\to\infty} \sum_{m\in\mathscr{M}} w_m F(m), \text{ where } F(m) = \widehat{f}_m.$$

From a practical point of view, the Markov chain $(M_t)_t$ can be generated as follows. We start from the transition $\Gamma(m, m')$ which is the transition probability of an aperiodic irreductible random walk on $\mathscr{M}$ and then implement the following algorithm.

Algorithm 1: Metropolis-Hastings

Data: a transition probability Γ, an arbitrary $M_1 \in \mathscr{M}$, a burn-in time T_0, $\{U_t\}_{t=T_0+1}^{T} \sim \mathrm{Unif}([0,1]^{T-T_0-1})$.

Result: $\frac{1}{T-T_0}\sum_{t=T_0+1}^{T} F(M_t)$.

for $t = 1, \ldots, T$ **do**

- From the current state M_t generate M'_{t+1} according to the distribution $\Gamma(M_t, \cdot)$;
- $p_{t+1} \leftarrow 1 \wedge \frac{w_{M'_{t+1}}\Gamma\left(M'_{t+1}, M_t\right)}{w_{M_t}\Gamma\left(M_t, M'_{t+1}\right)}$;

if $p_{t+1} < U_t$ *(i.e. with probability* p_{t+1} *do ...)* **then**

$M_{t+1} \leftarrow M'_{t+1}$.

end

else

$M_{t+1} \leftarrow M_t$.

end

end

The following example shows how to implement the Metropolis-Hastings algorithm for evaluating (1) in the coordinate sparse setting.

Example 4.1. We consider the collection of estimators $\left\{\widehat{f}_m : m \in \mathscr{M}\right\}$ and the probability π for the coordinate sparse setting. We write $\mathscr{V}(m)$ for the set of all the subsets m' that can be obtained from m by adding or removing one integer $j \in \{1, \ldots, p\}$ to m. If we take the uniform distribution on $\mathscr{V}(m)$ as proposal distribution $\Gamma(m, \cdot)$, then $\Gamma(m, m') = \Gamma(m', m)$ for any $m' \in \mathscr{V}(m)$. As a consequence, we only need to compute $w_{M'_{t+1}}/w_{M'_t}$ at the second step of the Metropolis Hastings algorithm. We have seen that

$$\frac{w_{m'}}{w_m} = \frac{\pi_{m'} e^{-\beta \widehat{r}_{m'}/\sigma^2}}{\pi_m e^{-\beta \widehat{r}_m/\sigma^2}}$$

so we need to compute $\pi_{m'}/\pi_m$ efficiently when $m' \in \mathscr{V}(m)$. Let us consider the choice $\pi_m \propto e^{-|m|}/C_p^{|m|}$ proposed for the coordinate-sparse setting. We have the simple formulas

$$\frac{\pi_{m'}}{\pi_m} = \frac{|m|+1}{e(p-|m|)} \text{ for } m' = m \cup \{j\} \quad \text{and} \quad \frac{\pi_{m'}}{\pi_m} = \frac{e(p-|m|+1)}{|m|} \text{ for } m' = m \backslash \{j\}.$$

5 Simulations [2]

We will consider the following Gaussian white noise model:

$$Y = f + \varepsilon, \text{ where } f(x) = x^5 - 8x^3 + 10x + 6, \quad \varepsilon \sim N(0, \sigma^2 I_n).$$

We suppose, that σ^2 is known and equals to 4.

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.random import default_rng

rng = default_rng()
n = 100
f = lambda x: x**5 - 8*x**3 + 10*x + 6
seq = np.linspace(-2.5, 2.5, n)
sig2 = 4
eps = rng.normal(0, sig2, size=n)
Y = f(seq) + eps
```

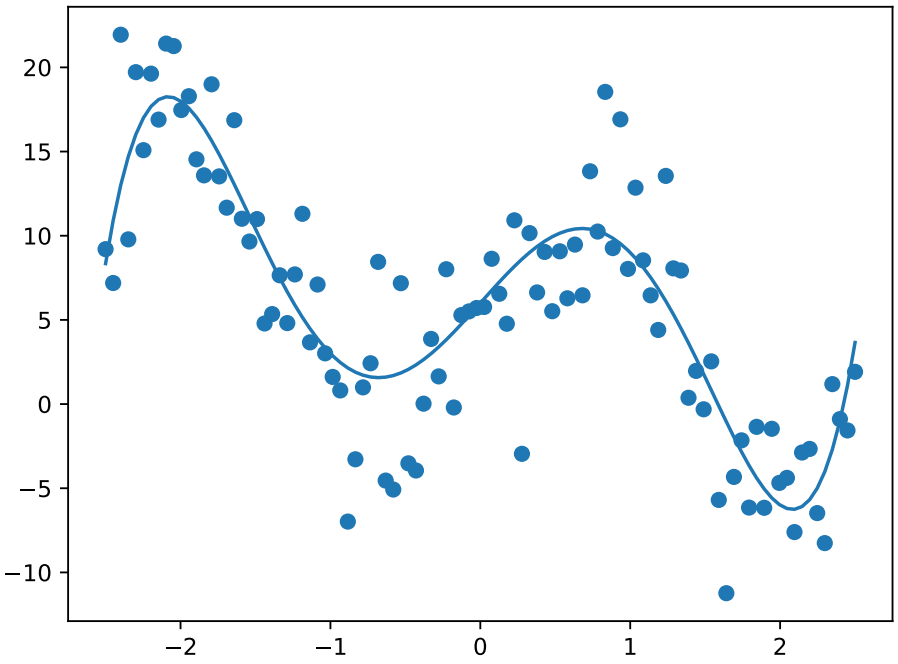

Figure 1: The simulated data Y (blue dots) and the unknown signal f (blue line).

We want to estimate this signal by expanding the observations on the Fourier basis.

$$\begin{cases} \varphi_1(x) = \dfrac{1}{\sqrt{P}}, & (7) \\ \varphi_{2k}(x) = \sqrt{\dfrac{2}{P}} \cos\left(\dfrac{2\pi k x}{P}\right), & (8) \\ \varphi_{2k+1}(x) = \sqrt{\dfrac{2}{P}} \sin\left(\dfrac{2\pi k x}{P}\right), \text{ where } P \text{ is a period.} & (9) \end{cases}$$

```python
# Fourier basis matrix (n_fct is odd)
def create_fourier_matrix(seq, n_fct):
  n = len(seq)
  basis_matrix = np.zeros((n, n_fct))
  basis_matrix[:, 0] = np.ones(n) / np.sqrt(100)

  for k in range(int((n_fct-1)/2)):
    basis_matrix[:, 2*k+1] = np.sin(2 * np.pi * (k+1) * seq / 5) / np.sqrt(50)
    basis_matrix[:, 2*k+2] = np.cos(2 * np.pi * (k+1) * seq / 5) / np.sqrt(50)

  return(basis_matrix)
```

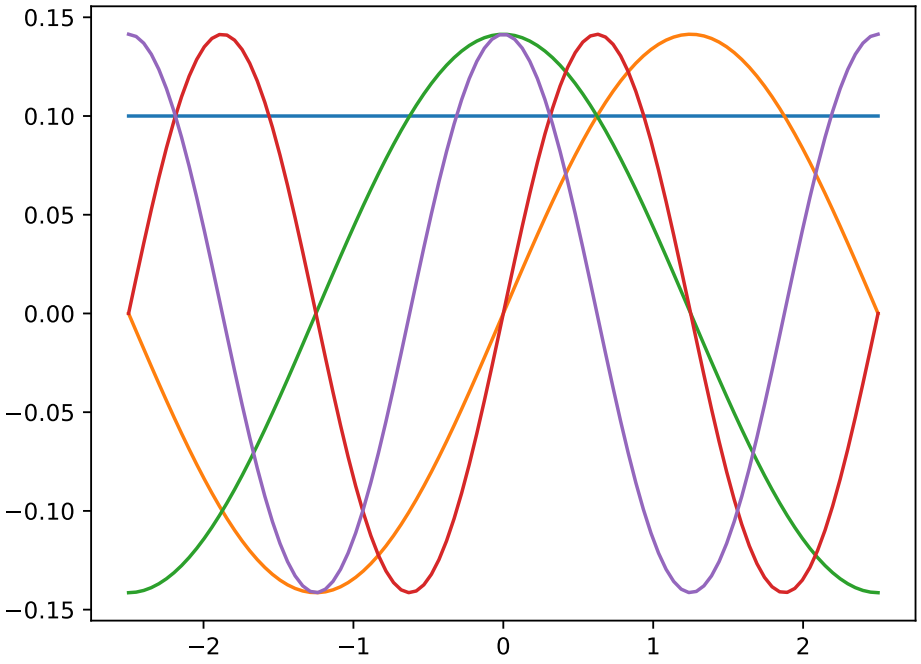

Figure 2: The first 5 functions of Fourier basis

The matrix $X \in \mathbb{R}^{n \times p}$ can be considered as orthogonal, because $X^T X \approx I_p$.

```python
X = create_fourier_matrix(seq, 9)
np.round(X.T @ X, 2)

'''array([[ 1.  ,  0.  , -0.01,  0.  ,  0.01,  0.  , -0.01,  0.  ,  0.01],
          [ 0.  ,  0.99, -0.  ,  0.  , -0.  ,  0.  , -0.  ,  0.  ,  0.  ],
          [-0.01, -0.  ,  1.01,  0.  , -0.02,  0.  ,  0.02, -0.  , -0.02],
          [ 0.  ,  0.  ,  0.  ,  0.99,  0.  , -0.  ,  0.  , -0.  ,  0.  ],
          [ 0.01, -0.  , -0.02,  0.  ,  1.01, -0.  , -0.02,  0.  ,  0.02],
          [ 0.  ,  0.  ,  0.  , -0.  , -0.  ,  0.99,  0.  ,  0.  ,  0.  ],
          [-0.01, -0.  ,  0.02,  0.  , -0.02,  0.  ,  1.01, -0.  , -0.02],
          [ 0.  ,  0.  , -0.  , -0.  ,  0.  ,  0.  , -0.  ,  0.99,  0.  ],
          [ 0.01,  0.  , -0.02,  0.  ,  0.02,  0.  , -0.02,  0.  ,  1.01]])'''
```

Now we will calculate the coefficients Z of Y in the Fourier basis: $Z = X^T Y$, $Z \sim N(\theta, \sigma^2 I_p)$. Analysing the positioning of Z with respect to the lines $y = \pm\sqrt{2\sigma^2 \log(p)}$, we can consider the expectation θ as sparse vector.

```
p = 101
X = create_fourier_matrix(seq, p)
Z = X.T @ Y
plt.scatter(np.linspace(-2.5, 2.5, len(Z)), Z)
plt.axline((0, np.sqrt(2 * sig2 * np.log(p))), (1, np.sqrt(2 * sig2 * np.log(p)
    ))), color='r')
plt.axline((0, -np.sqrt(2 * sig2 * np.log(p))), (1, -np.sqrt(2 * sig2 * np.log(
    p))), color='r')
plt.show()
```

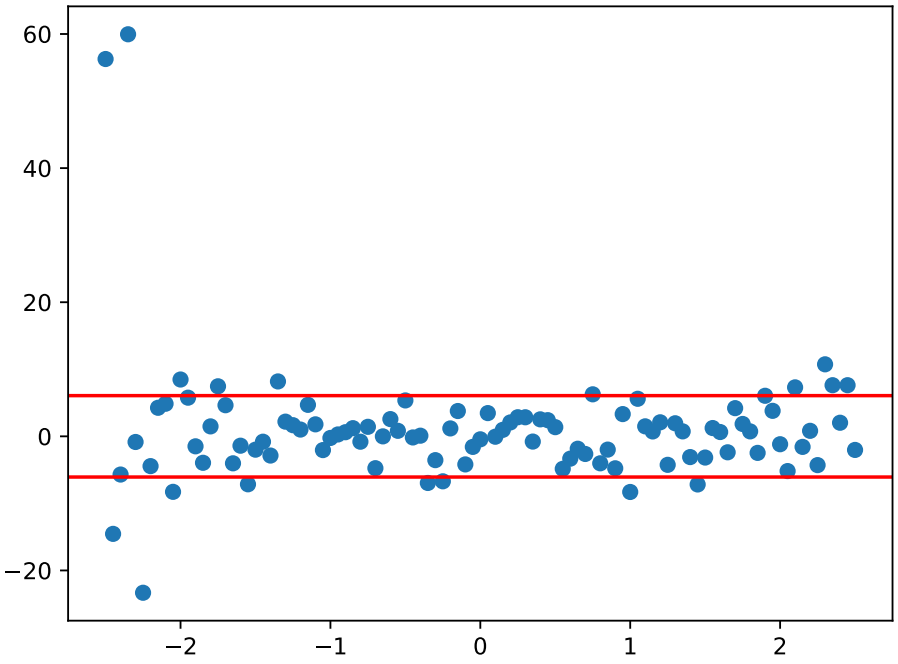

Figure 3: Coefficients Z of Y in the Fourier basis

We implemented the thresholding estimator for θ:

$$\widehat{\theta}_j = Z_j \mathbb{1}_{\{|Z_j| \geq \sqrt{2\sigma^2 \log(p)}\}}, \text{ for } j = 1, \ldots, p.$$

```
n_models = 50
error = []
for k in range(n_models):
  X = create_fourier_matrix(seq, (2*k+1))
  Z = X.T @ Y
  p = 2*k+1
  t = np.sqrt(2 * sig2 * np.log(p))
  theta_hat = Z * (np.abs(Z) >= t)
  f_hat = X @ theta_hat
```

```
  error.append(np.sum((f_hat - f(seq))**2))

k_opt = 2*(error.index(min(error))+1)+1
print("m_hat =", k_opt)
X = create_fourier_matrix(seq, k_opt)
Z = X.T @ Y
p = k_opt
t = np.sqrt(2 * sig2 * np.log(p))
theta_hat = Z * (np.abs(Z) >= t)
f_hat = X @ theta_hat
plt.scatter(seq, Y)
plt.plot(seq, f(seq))
plt.plot(seq, f_hat, color='orange')
print("Error =", np.round(np.sum((f_hat - f(seq))**2), 2))
```

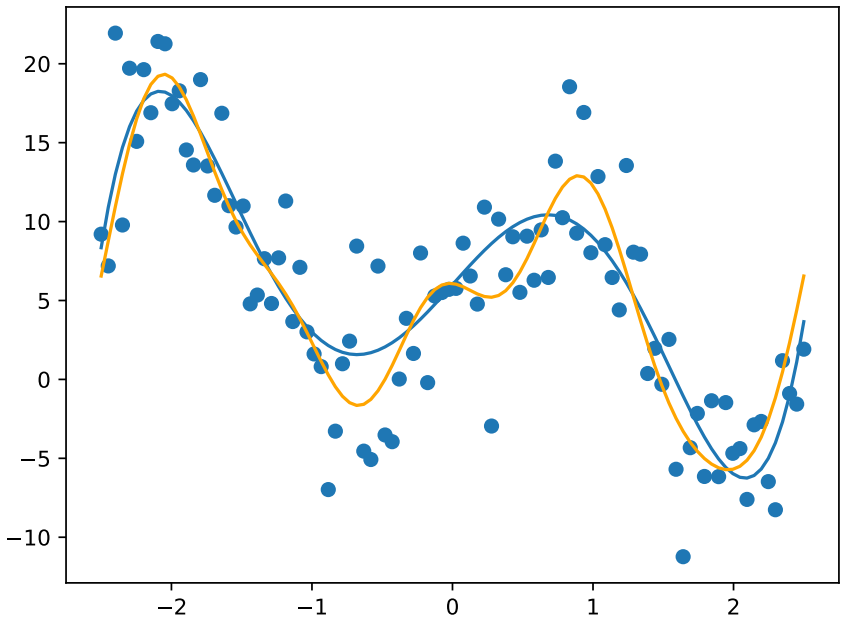

Figure 4: The selection model with the best oracle risk, error = 357.04.

Probably, the choice of another basis could have given better results.

We take all π_m's equal and select $\widehat{m}$ according to the criterion (3).

```
n_models = 50
K = 1.1
error = []
for k in range(n_models):
  X = create_fourier_matrix(seq, (2*k+1))
  Z = X.T @ Y
  p = 2*k+1
  t = np.sqrt(2 * sig2 * np.log(p))
  theta_hat = Z * (np.abs(Z) >= t)
  f_hat = X @ theta_hat
```

```
  m_hat = np.sum((f_hat - Y)**2) + sig2 * K * (1 + np.sqrt(2 * np.log(n_models)
   ))**2
  error.append(m_hat)

k_opt = 2*(error.index(min(error))+1)+1
print("m_hat =", k_opt)
X = create_fourier_matrix(seq, k_opt)
Z = X.T @ Y
p = k_opt
t = np.sqrt(2 * 4 * np.log(p))
theta_hat = Z * (np.abs(Z) >= t)
f_hat = X @ theta_hat
print("Error =", np.round(np.sum((f_hat - f(seq))**2), 2))
```

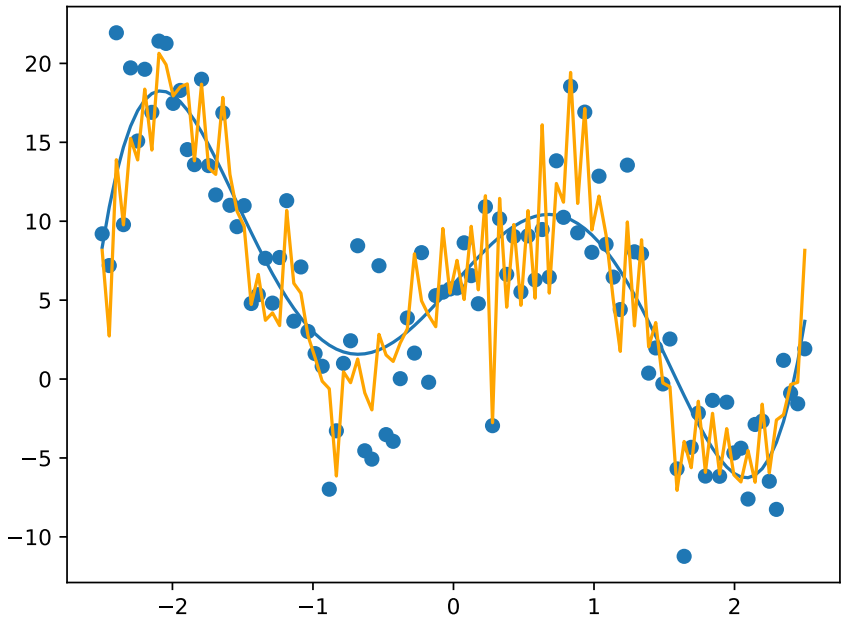

Figure 5: The selection model $\widehat{m}$, error = 1079.9. Overfitting.

Now let us try to approximate F by Metropois-Hastings algorithm. $\Gamma(M_t, \cdot)$ is uniform on M.

```
n_models = 50
# A set of estimators f_hat_set and a set of risks r_hat_set
f_hat_set = np.array([])
r_hat_set = np.array([])
for k in range(n_models):
  X = create_fourier_matrix(seq, (2*k+1))
  Z = X.T @ Y
  p = 2*k+1
  t = np.sqrt(2 * 4 * np.log(p))
  theta_hat = Z * (np.abs(Z) >= t)
  f_hat = X @ theta_hat
  f_hat_set = np.append(f_hat_set, f_hat)
  dim = 2 * k + 1
  r_hat = np.sum((Y-f_hat)**2) + 2 * dim * sig2 - n * sig2
```

```
  r_hat_set = np.append(r_hat_set, r_hat)
f_hat_set = f_hat_set.reshape(n_models, n)

T_0 = 0
T = 10000
M_t = 4
U = rng.uniform(size=T)
F = np.zeros(n)
beta = 1/4

for t in range(T-T_0):
  M_prime = rng.choice(n_models, 1) # uniform choice of M'_{t+1} from M
  if M_prime - M_t == 1:
    ratio_pi = (M_t + 1) / (np.exp(1) * (n_models - M_t))
  elif M_t - M_prime == 1:
    ratio_pi = (np.exp(1) * (n_models - M_t + 1)) / M_t
  else:
    ratio_pi = 1  # for easier calculations
  ratio_w = ratio_pi * np.exp(beta / sig2 * (r_hat_set[M_prime] - r_hat_set[M_t
   ]))
  p_t = np.min([1, ratio_w])
  if p_t < U[t]:
    M_t = M_prime
  else:
    M_t = M_t
  F = F + f_hat_set[M_t, :] / (T-T_0)
```

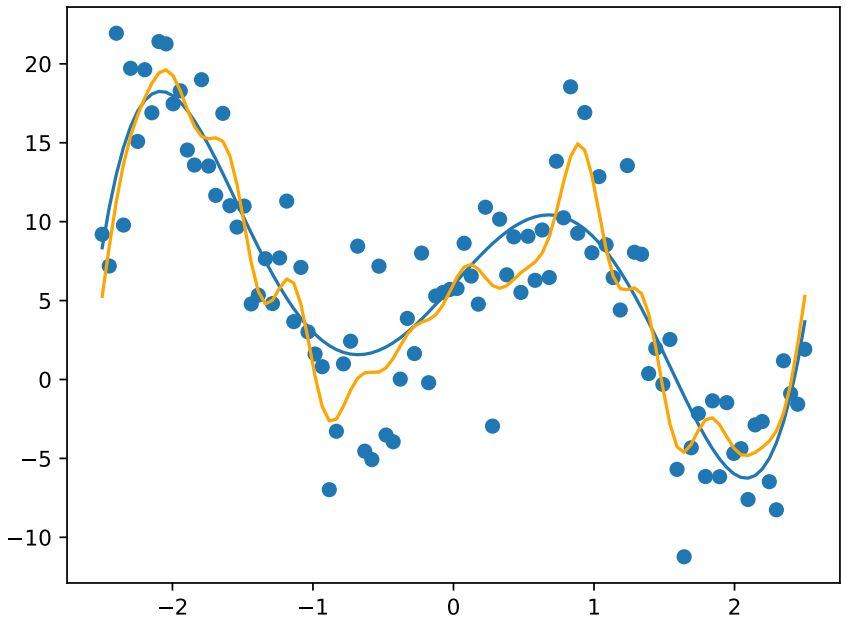

Figure 6: The model with aggregated estimator $\widehat{f}$, error = 437.85.

As we can see, on this type of data aggregated estimator indeed performs better than the selected one.

References

[1] Christophe Giraud. *Introduction to High-Dimensional Statistics*. Universite Paris Saclay, 2021.

[2] *Google colab: Aggregation of estimators.ipynb*. URL: https://colab.research.google.com/drive/1iWqtSil1MifM6GlW-rRaVHvZop8aEPdZ?usp=sharing.